Statements Grouped by Functions

This section provides an overview of the statements grouped by their functions.

This section covers the following topics:

- Natural SQL Statements - Overview
- Database Access and Update
- Arithmetic and Data Movement Operations
- Loop Execution
- Creation of Output Reports
- Screen Generation for Interactive Processing
- Processing of Logical Conditions
- Invoking Programs and Routines
- Control of Work Files
- Component Based Programming
- Event-Driven Programming
- Miscellaneous

Database Access and Update

The following statements are used to access and manipulate information contained in a database.

READ	Reads a database file in physical or logical sequence of records.
FIND	Selects records from a database file based on user-specified criteria.
HISTOGRAM	Reads the values of a database field.
GET	Reads a record with a given ISN (internal sequence number) or RNO (record number).
GET SAME	Re-reads the record currently being processed.
ACCEPT/REJECT	Accepts/reject records based on user-specified criteria.
PASSW	Provides password for access to a password-protected file.
LIMIT	Limits the number of executions of a READ, FIND or HISTOGRAM processing loop.
STORE	Adds a new record to the database.
UPDATE	Updates a record in the database.
DELETE	Deletes a record from the database.
END TRANSACTION	Indicates the end of a logical transaction.
BACKOUT TRANSACTION	Backs out a partially completed logical transaction.
GET TRANSACTION DATA	Reads transaction data stored with a previous END TRANSACTION statement.
RETRY	Attempts to re-read a record which is in hold status for another user.
AT START OF DATA	Specifies statements to be performed when the first of a set of records is processed in a processing loop.
AT END OF DATA	Specifies statements to be performed after the last of a set of records has been processed in a processing loop.
AT BREAK	Specifies statements to be performed when the value of a control field changes (break processing).
BEFORE BREAK PROCESSING	Specifies statements to be performed before performing break processing.
PERFORM BREAK PROCESSING	Immediately invokes break processing.

Arithmetic and Data Movement Operations

The following statements are used for arithmetic and data movement operations:

COMPUTE	Performs arithmetic operations or assigns values to fields.
ADD	Adds two or more operands.
SUBTRACT	Subtracts one or more operands from another operand.
MULTIPLY	Multiplies two or more operands.
DIVIDE	Divides one operand into another.
MOVE	Moves the value of an operand to one or more fields.
MOVE ALL	Moves multiple occurrences of a value to another field.
COMPRESS	Concatenates the value of two or more fields into a single field.
SEPARATE	Separates the content of a field into two or more fields.
EXAMINE	Scans a field for a specific value and replaces it, and/or counts how often it occurs.
RESET	Sets the value of a field to zero (if numeric) or blank (if alphanumeric), or to its initial value.

Loop Execution

The following statements are related to the execution of processing loops:

REPEAT	Initiates a processing loop (and terminates it based on a specified condition).
FOR	Initiates a processing loop and controls the number of times the loop is to be processed.
ESCAPE	Stops the execution of a processing loop.

Creation of Output Reports

The following statements are used for the creation of output reports:

FORMAT	Specifies output parameter settings.
DISPLAY	Specifies fields to be output in column form.
WRITE/PRINT	Specifies fields to be output in non-column form.
WRITE TITLE	Specifies text to be output at the top of each page of a report.
WRITE TRAILER	Specifies text to be output at the bottom of each page of a report.
AT TOP OF PAGE	Specifies processing to be performed when a new output page is started.
AT END OF PAGE	Specifies processing to be performed when the end of an output page is reached.
SKIP	Generates one or more blank lines in a report.
EJECT	Causes a page advance without titles or headings.
NEWPAGE	Causes a page advance with titles and headings.
SUSPEND IDENTICAL SUPPRESS	Suspends identical suppression for a single record.
DEFINE PRINTER	Allocates a report to a logical output destination.
CLOSE PRINTER	Closes a printer.

Screen Generation for Interactive Processing

The following statements are used to create data screens (maps) for the purpose of interactive processing of data:

INPUT	Creates a formatted screen (map) for data display/ entry.
REINPUT	Re-executes an INPUT statement (if invalid data were entered in response to the previous INPUT statement).
DEFINE WINDOW	Specifies the size, position and attributes of a window.
SET WINDOW	Activates and de-activates a window.

Processing of Logical Conditions

The following statements are used to control the execution of statements based on conditions detected during the execution of a Natural program:

IF	Performs statements depending on a logical condition.
IF SELECTION	Verifies that in a sequence of alphanumeric fields one and only one contains a value.
DECIDE FOR	Performs statements depending on logical conditions.
DECIDE ON	Performs statements depending on the contents of a variable.
ON ERROR	Intercepts runtime errors which would otherwise result in a Natural error message, followed by the termination of the Natural program.

Invoking Programs and Routines

The following statements are used in conjunction with the execution of programs and routines:

FETCH	Invokes a Natural program.
CALLNAT	Invokes a Natural subprogram.
PERFORM	Invokes a Natural subroutine.
DEFINE SUBROUTINE	Defines a Natural subroutine.
ESCAPE	Stops the execution of a routine.
CALL	Invokes a non-Natural program from a Natural program.
CALL FILE	Invokes a non-Natural program to read a record from a non-Adabas file.
CALL LOOP	Generates a processing loop containing a call to a non-Natural program.

Control of Work Files

The following Natural statements are used to read/write data to a physical sequential (non-Adabas) work file:

WRITE WORK FILE	Writes data to a work file.
READ WORK FILE	Reads data from a work file.
CLOSE WORK FILE	Closes a work file.
DEFINE WORK FILE	Assigns a file name to a work file.

Component Based Programming

The following Natural statements are used in conjunction with component based programming:

DEFINE CLASS	Specifies a class from within a Natural class module.
CREATE OBJECT	Creates an object (also known as an instance) of a given class.
SEND METHOD	Invokes a method of an object.
INTERFACE	Defines an interface (a collection of methods and properties) for a certain feature of a class.
METHOD	Assigns a subprogram as the implementation of a method, outside an interface definition.
PROPERTY	Assigns an object data variable as the implementation to a property, outside an interface definition.

Event-Driven Programming

The following Natural statements are used for event-driven programming:

OPEN DIALOG	Opens a dialog.
CLOSE DIALOG	Closes a dialog.
SEND EVENT	Triggers a user-defined event.
PROCESS GUI	Performs a standard procedure in an event-driven application.

Miscellaneous

DEFINE DATA	Defines the data elements which are to be used in a Natural program or routine.
END	Indicates the end of the source code of a Natural program or routine.
EXPAND	Expands the allocated memory of dynamic variables to a given size.
EXAMINE TRANSLATE	Translates the characters contained in a field into upper-case or lower-case, or into other characters.
INCLUDE	Incorporates Natural copycode at compilation.
PROCESS COMMAND	Invokes a command processor.
REDUCE	Reduces the allocated memory of dynamic variables.
RELEASE	Deletes the contents of the Natural stack; releases sets of ISNs/RNOs retained via a FIND statement; releases Natural global variables.
REQUEST DOCUMENT	Allows you to access an external system.
RUN	Compiles and executes a source program.
SET CONTROL	Performs a Natural terminal command from within a Natural program.
SET KEY	Assigns functions to terminal keys.
SETTIME	Establishes a point-in-time reference for a *TIMD system variable.
SORT	Sorts records, using the sort program provided by the operating system.
STACK	Places data and/or commands into the Natural stack.
STOP	Terminates the execution of an application.
TERMINATE	Terminates the Natural session.

